

17. JULI 2015



der **BÜRO**Führer

PROJEKTARBEIT

GESELLENSTÜCK

PATRICK.KRICK

Inhalt

1.Die Idee	2
2.Die Hardware	3
2.1.AVR Controller	3
2.2.Bewegungsmelder.....	3
2.3.Kamera	4
2.4.Servo Motor	4
3.Der Bau des Gesellenstücks	5
4.Softwareentwicklung des AVR Controllers.....	13
Quellenverzeichnis	16

1. Die Idee

Da die Zeit zu meiner Gesellenprüfung immer näher rückte und ich noch keine Idee für mein Gesellenstück hatte, hörte ich mir ein paar Vorschläge von Arbeitskollegen an. Nach mehreren Ratschlägen war ich aber immer noch nicht weiter, was das Thema betraf. Tatsächlich ist die Idee ebenso schnell wie einfach entstanden. Als wir uns freitags nachmittags in der Firma unterhielten, erfuhr ich, dass ein Kollege seine private Hausüberwachung aufrüstete.

Da ist mir die Idee gekommen eine Kamerasteuerung mit Hilfe des AVR Controllers zu realisieren. In den nächsten Tagen habe ich mir noch Gedanken über den genauen Aufbau gemacht und wie ich dies am besten umsetzen kann. Letzten Endes entstand dann ein Prototyp in meinem Kopf, mit zwei Bewegungsmeldern und einer Kamera, die sich mit Hilfe eines Servomotors dann nach links oder rechts dreht, wenn einer der Bewegungsmelder anschlägt.

Anschließend ging es darum, welche Bauteile am besten dafür geeignet sind. Ich setzte mir als Vorgabe das Ganze möglichst klein zu bauen und suchte dann bei verschiedenen Anbietern nach den passenden Teilen.

2. Die Hardware

Ich habe mich für folgende Hardware entschieden, welche ich in den nächsten Punkten genauer erklären werde.

2.1. AVR Controller

Bereits in der Zwischenprüfung haben wir im praktischen Teil ein AVR Board mit Atmega Controller zusammengelötet. Diesen wollte ich nicht außer Acht lassen und habe mich dazu entschlossen ihn für das Gesellenstück zu verwenden.

Zur Funktion des Microcontrollers: Er soll die Signale der Bewegungsmelder auslesen und dann je nach Anschlag den Servomotor mit einer Impulsbreitenregelung in die jeweilige Richtung drehen lassen. Nach einer gewissen Zeit ohne Signal soll dann der Servomotor in die Nullstellung zurück fahren.



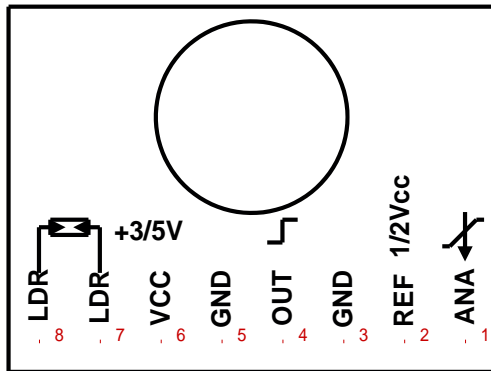
2.2. Bewegungsmelder

Ich habe mich für einen Passiv Infrarot Bewegungsmelder in einer Low Power Ausführung entschieden. Der PIR-LP ist optimal dafür geeignet, da er erstens sehr klein ist und zweitens durch einen Fensterkomperator ein digitales Schaltsignal abgibt, das gut von einem Mikrocontroller weiterverarbeitet werden kann.



Die Funktion des Bewegungsmelders ist wie folgt: erst wenn die Teilsegmente des Sensors mit verschiedenen Infrarotstrahlen beaufschlagt werden, ändert sich das Ausgangssignal. Das heißt, wenn eine Person durch den Raum läuft bekommt

B1 ein oder mehrere Segmente eine andere Infrarotstrahlung ab, als **IR PIR-LP 6235** die anderen.



Die Pin Belegung der Bewegungsmelder ist wie folgt.

2.3.Kamera

Zur Kamera gibt es nichts Besonderes zu sagen. Es ist eine normale Webcam die ich via USB an meinen Laptop anschließen kann. Ich habe lediglich darauf geachtet, dass die Kamera eine praktische Halterung zum Befestigen auf dem Servomotor hat.



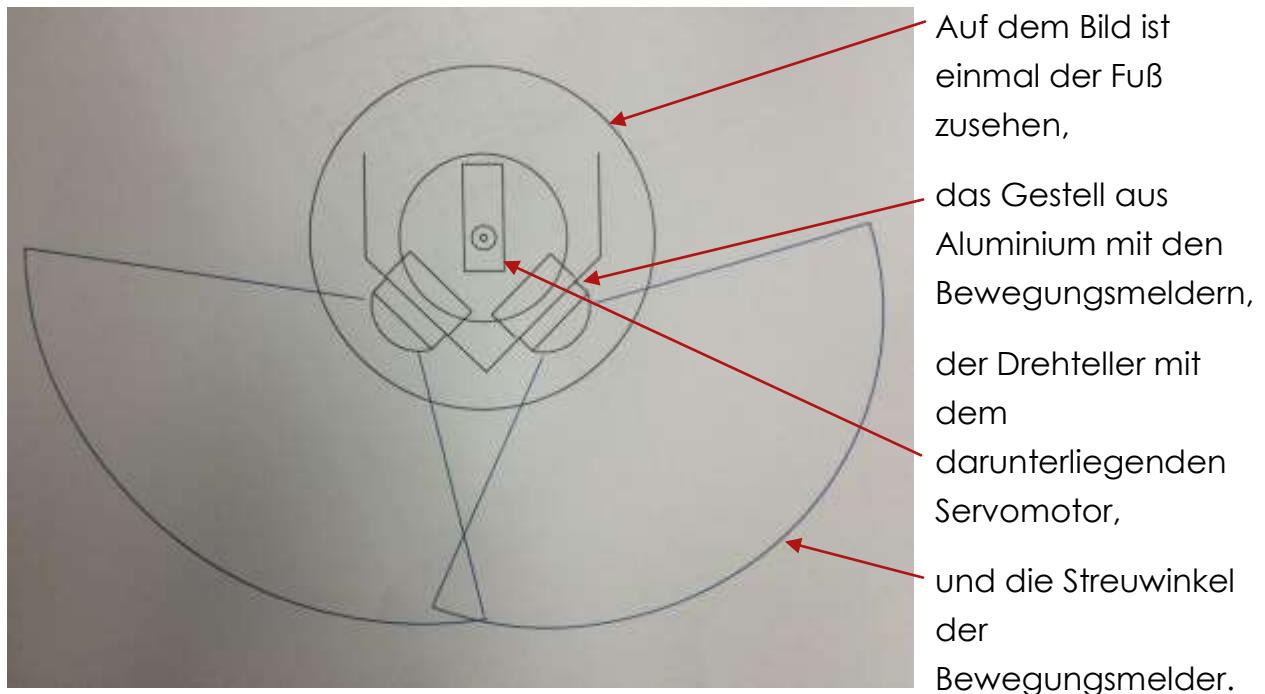
2.4.Servo Motor

Der Servomotor MC-1811 von Modelcraft wird mit einer variablen Impulsbreite betrieben. Man kann sich das in etwa so vorstellen, dass jede „Stellung“ des Servomotors eine bestimmte Impulsbreite hat. Möchte man ihn also auf 50% des maximalen Drehwinkels stellen, so muss man eine Impulsbreite von 50% der Vollansteuerung auf den Servomotor geben.

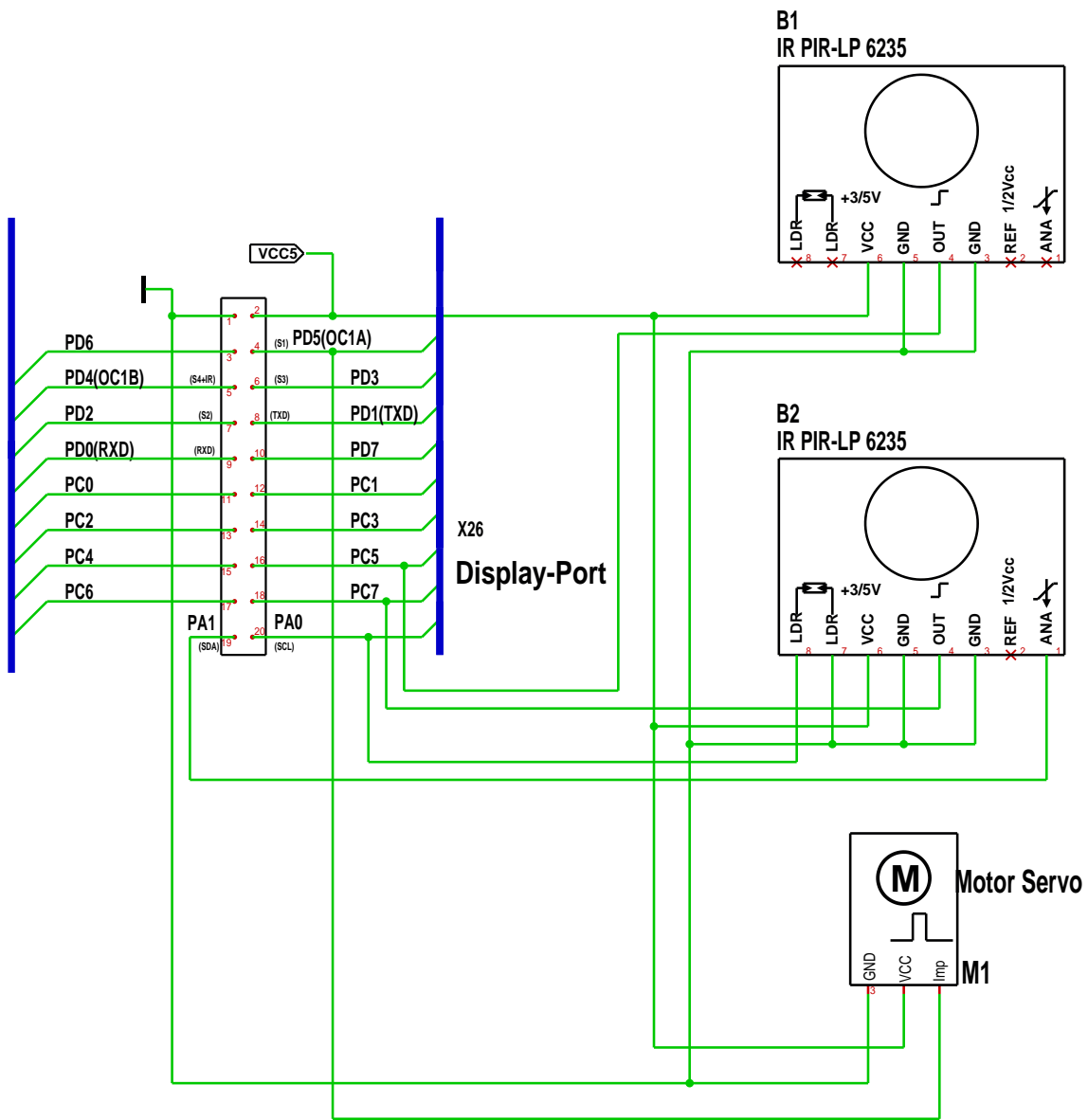


3. Der Bau des Gesellenstücks

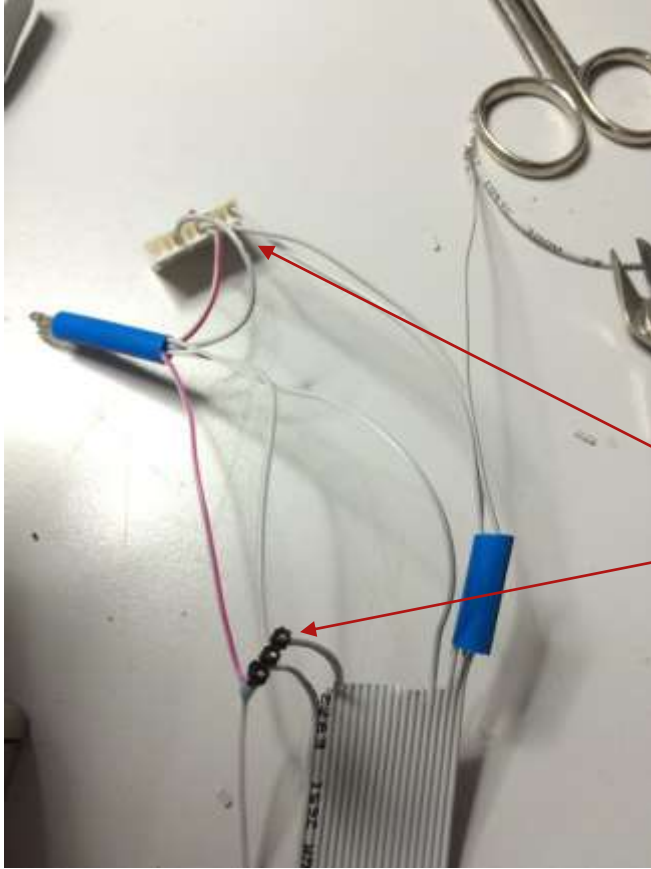
Ich überlegte zunächst, wie ich meine Idee der Kamerasteuerung sehr kompakt in ein relativ stabiles Gestell bauen kann und kam dann zu dem Entschluss ein Blech zu biegen, daran dann in einem 90° Winkel die Bewegungsmelder zu befestigen und oben drauf den Servomotor mit Kamera zu montieren. Also begann ich mit einer kleinen Zeichnung von oben.



Danach widmete ich mich aber zunächst der Elektronik. In Zusammenarbeit mit Herr Bork habe ich einen Schaltplan entworfen um die Verkabelung klar zu stellen. In diesem Zug habe ich auch gleich das Flachbandkabel angefertigt. Ich habe einfach laut Schaltplan gehandelt und jeden Anschluss so gelegt dass am Ende jede Ader an der richtigen Stelle ist.



Maßstab	77,00%	Datei	Kamera Steuerung mit AVRKit	Zeichner	P.Krick	Blatt
Anderung	07.07.15	15:46		Titel		
Ausgabe	07.07.15	19:54		Kamera Steuerung mit AVRKit		
Firma	Der Büroführer			Projekt	Seib AVR-Kit	16



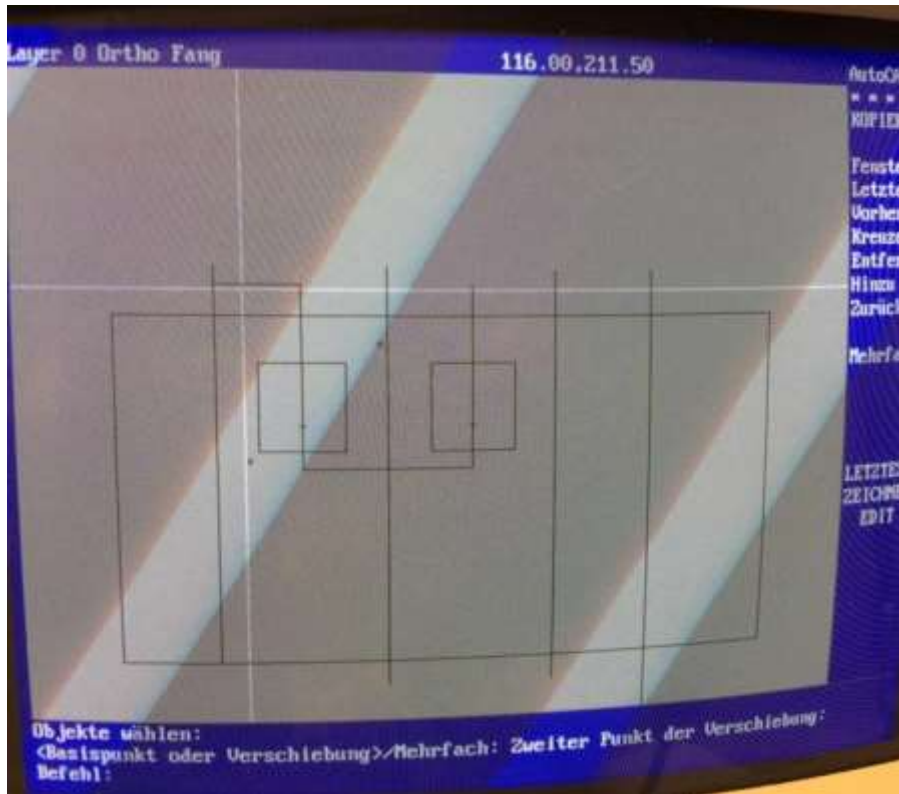
Die Herstellung des Kabels war kein Problem (auf dem Bild ist nur ein Zwischenstadium zu sehen). Ich habe alle Erdleiter und Spannungsversorger zusammengefasst. Dann habe ich die Kabel, die bei den Bewegungsmeldern gebraucht werden in ein passendes Verbindungsstück eingehakt. Danach habe ich noch ein Verbindungsstück für den Servomotor gebaut, damit auch dieser angesteuert werden kann.

Jetzt ging es an das Gestell. Ich hatte Glück, dass ich die Werkstatt von Herrn Bork nutzen durfte, an dieser Stelle vielen Dank dafür.

Da ich die Fräse nicht eigenständig programmieren konnte, entwarf ich mit Hilfe von Herrn Bork das Layout vom Gestell, dem Fuß und dem Drehteller. Wir haben uns für Aluminium als Material entschieden, da dies relativ einfach zu bearbeiten ist. Der Drehteller wurde für schönere Optik aus Plexiglas hergestellt und den Fuß fertigten wir ebenfalls in der Fräse aus einer einfachen Holzplatte. Das Gestell benötigte noch Aussparungen für die zwei Bewegungsmelder und den Servomotor.

Die Bilder folgen in einer Fotoserie.

Layout Entwurf für das Aluminiumgestell.



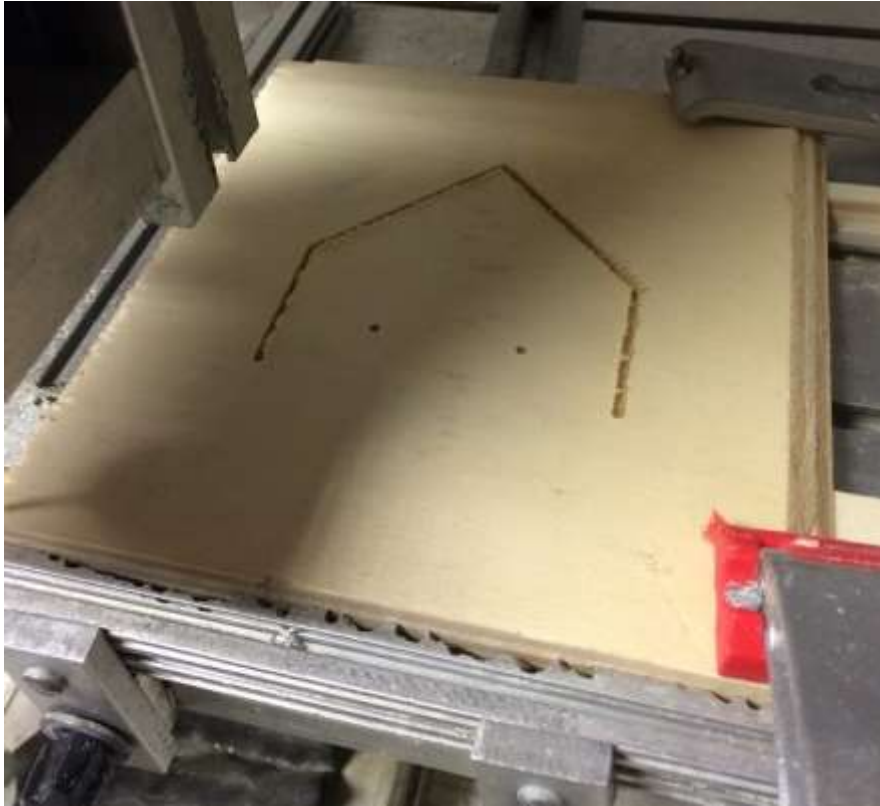
Fräsen des Gestells.
Zu sehen sind die
Ausparungen für
die
Bewegungsmelder





So sieht das fertige, noch unbestückte Gestell aus. Nachdem es gefräst wurde, habe ich es einfach im richtigen Winkel gebogen und die Halterung für den Servomotor daran verschraubt. Danach wurde der Fuß gefräst und mit einer Versenkung für das Gestell versehen.

Zu sehen ist der Fräsvorgang vom Fuß. Die Versenkung ist bereits fertig.



Die Herstellung des Drehtellers habe ich leider nicht fotografiert, sie ist aber ähnlich wie der Rest entstanden.

Nachdem dann alle Teile fertig waren, baute ich die Hardware ein. Ich befestigte die Bewegungsmelder mit 4 einfachen Heißklebestellen an der Innenseite.

Um den Servomotor zu befestigen, haben wir beim Fräsen kleine Löcher zum verschrauben vorgesehen.

Der Drehteller wurde einfach auf den Servomotor geschraubt und die Kamera mit einer Klammer darauf befestigt.

Danach habe ich das zuvor gefertigte Anschlusskabel in die richtigen Bauelemente gesteckt und fertig war die Hardware.



4. Softwareentwicklung des AVR Controllers

Die Software des AVR Kits musste Stück für Stück stattfinden. Zunächst hat Herr Bork eine Testversion der Steuerung des Servomotors entworfen. Da der Servomotor mit einer Impulsbreite angetrieben wird, ist es wichtig zunächst einmal die richtige Impulsbreite zu finden, so dass der Servomotor den richtigen Drehwinkel hat.

Darauf möchte ich aber nicht genauer eingehen. Das Wichtigste am Programm ist die Erfassung der Signale der Bewegungsmelder, um diese dann in einen Winkelgrad umzuwandeln.

Am Anfang musste ich Variablen festlegen, um mein Programm zu vereinfachen. Welche ich genau brauche stellte sich natürlich im Laufe der Programmierung raus.

```
Dim ActionzeitBL as Single      'Namensgebung der Lowimpule Links
Dim ActionzeitBR as Single      'Namensgebung der Lowimpule Rechts
Dim Mittelwert as Single        'Prozentwert %
Dim Richtung as Boolean         'Links=1 Rechts=0
Dim Winkelmass as Single        'Winkelgrad °
```

Im Hauptprogramm muss zunächst erstmal gemessen werden, wie ,viele' Impulse von den Bewegungsmeldern gesendet werden, wenn sich jemand in einem Bereich bewegt. Dazu habe ich einen einfachen Zähler der Lowimpulse programmiert.

```
IF BL=0 Then Incr ActionzeitBL  'Zeiterfassung
IF BR=0 Then Incr ActionzeitBR
```

Diese Programmzeile macht nichts anderes, als die variable ,Actionzeit...' um Eins zu erhöhen und zwar immer dann, wenn der Bewegungsmelder auf null steht.

Zuvor habe ich den Maximalwert der Lowimpulse auf 600 limitiert, indem ich dann einfach den Zählvorgang gestoppt habe

```
INCR I
IF I = 600 Then
  I=0 :V4=1
Stop TIMER0
```

Danach haben wir festgelegt, dass er die zwei Werte, die er zuvor ermittelt hat vergleicht und dadurch die Richtung festlegt auf welcher Seite mehr Signale kommen.

```
IF ActionzeitBL>ActionzeitBR Then
  Richtung=1
Else
  Richtung=0
End if
```

Um dann den Winkelgrad zu errechnen, brauchten wir zunächst einen Mittelwert aus den beiden gemessenen Werten. Das habe ich mit einer einfachen Rechnung gemacht.

```
Mittelwert=ActionzeitBL+ActionzeitBR 'Mittelwert berechnen
Mittelwert=Mittelwert/2
```

Dann habe ich durch eine Rechnung den Faktor ermittelt um aus dem Mittelwert eine Zahl zwischen 0 und 100 zu bekommen, sodass ich mit dieser weiter arbeiten kann. Die errechnete Zahl gibt an, zu wie viel Prozent die Signale überwiegen. Anschließend habe ich aus der Prozentzahl eine Gradzahl berechnet, da die Kamera ja von 0° auf 180° dreht. Es war also nur notwendig aus den 100% einen Winkelgrad von 90° zu errechnen.

```
Mittelwert=Mittelwert*0.16666 'Prozentsatz ermitteln
Winkelmass=Mittelwert*0.9 'aus Prozent ein Winkelgrad berechnet
```

Als letztes hab ich eine Selektion der Richtungen programmiert, in der dann die Anweisung steht was gemacht werden soll.

```
Select case Richtung
case 0: Winkelmass=90-Winkelmass
case 1: Winkelmass=90+Winkelmass
End Select
```

Und so sieht das komplette Hauptprogramm aus.

```
BASCOM-AVR IDE [2.0.7.6] - [C:\Users\Patrick.Knick.DBFKH\Desktop\Projektarbeit\Bascom\PK Kamerasteuerung V2.bas]
Datei Editieren Anzeigen Programmieren Werkzeuge Optionen Fenster Hilfe
history.bt PK KamerasteuerungV2.bas
Code Explorer Sub Label
Aliases
Constants
Declarations
Labels
Subs
Variables
V5 = Not V6
'-----Torzeit berechnung des Winkelgrades-----
Print"BL: ";ActionzeitBL
Print"BR: ";ActionzeitBR
IF ActionzeitBL>ActionzeitBR Then
  Richtung=1
Else
  Richtung=0
End if
Mittelwert=ActionzeitBL+ActionzeitBR 'Mittelwert berechnen
Mittelwert=Mittelwert/2
Mittelwert=Mittelwert*0.16666 'zunächst Prozentsatz ermitteln
Winkelmass=Mittelwert*0.9 'aus Prozent ein Winkelgrad berechnet
Print"VorherGrad: ";Winkelmass
Select case Richtung
case 0: Winkelmass=90-Winkelmass
case 1: Winkelmass=90+Winkelmass
End Select
ActionzeitBL=0
ActionzeitBR=0
Call ServoWinkelGeber (Winkelmass)
Print"Grad: ";Winkelmass
wait 1
Start TIMERO
V4=0
End IF
162: 56 Modified Insert
```


Quellenverzeichnis

www.Conrad.de